



دانشگاه صنعتی امیرکبیر  
(پل تکنیک تهران)  
دانشکده ریاضی و علوم کامپیوتر

### درس داده کاوی

کار با مجموعه داده و مدل درخت تصمیم با کمک نرم افزار Rapid miner

نگارش:

علی‌اکبر گلستانی  
(۹۹۱۳۰۲۳)

معین رضایی  
(۹۷۱۲۰۱۹)

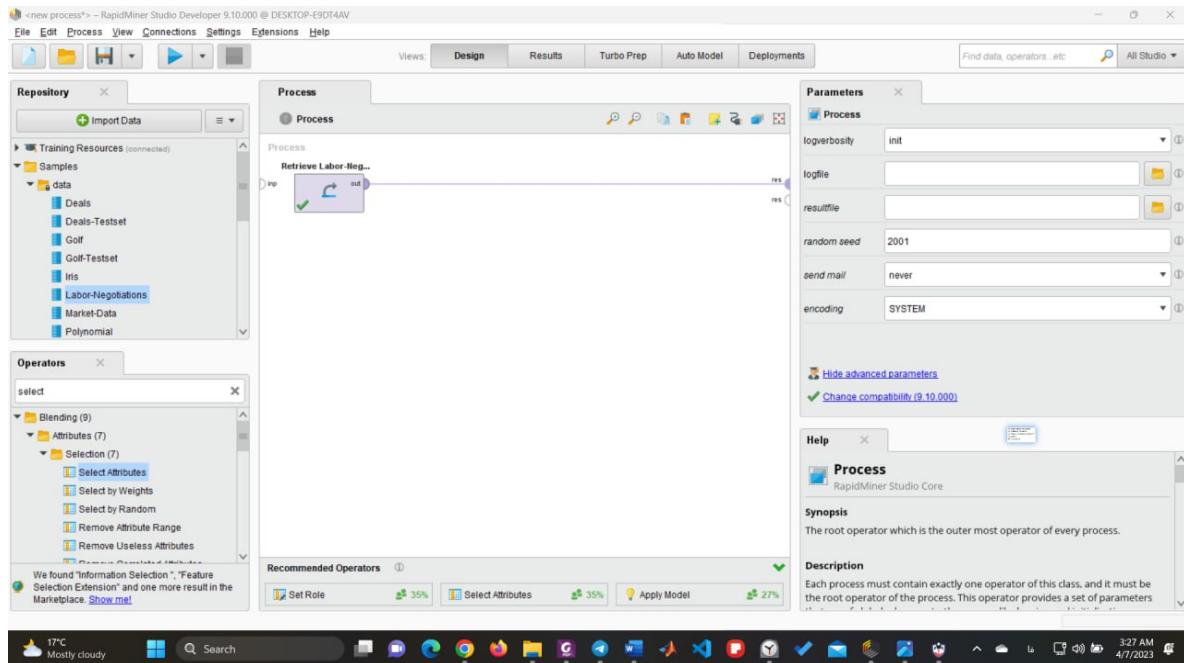
امیر آزاد  
(۹۸۱۳۳۰۱)

استاد درس:  
دکتر فاطمه شاکری

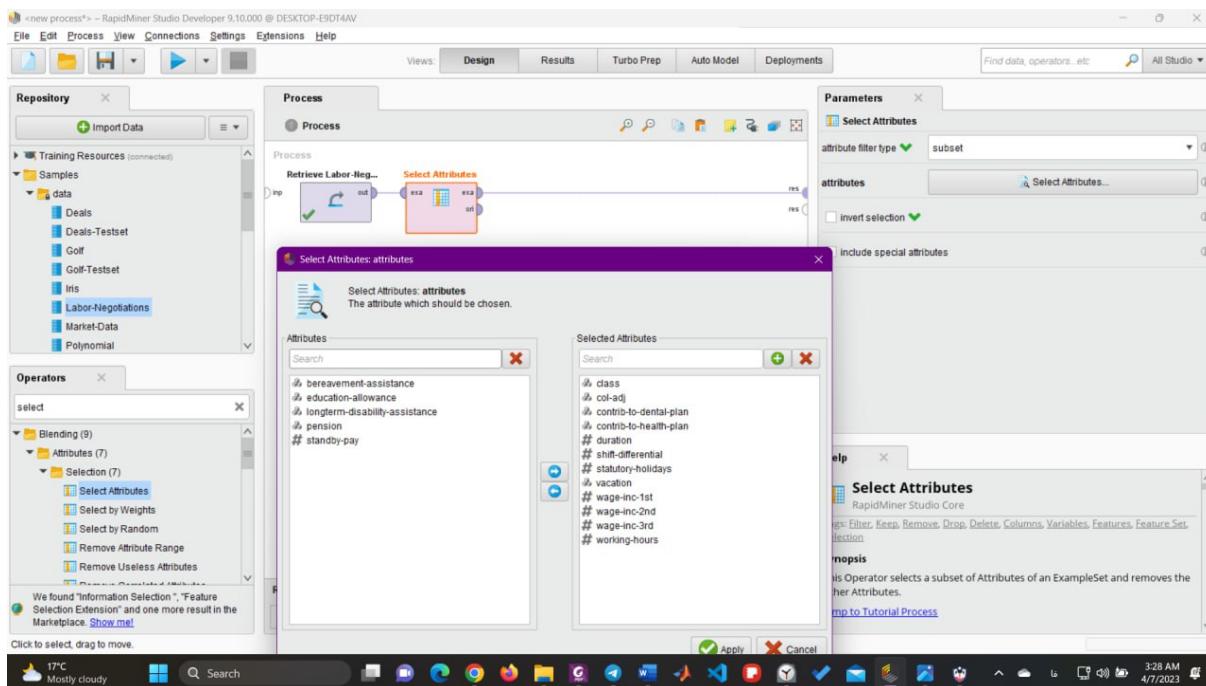
فروردین ۱۴۰۲

## بخش اول - کار با مجموعه داده Labor-Negotiations

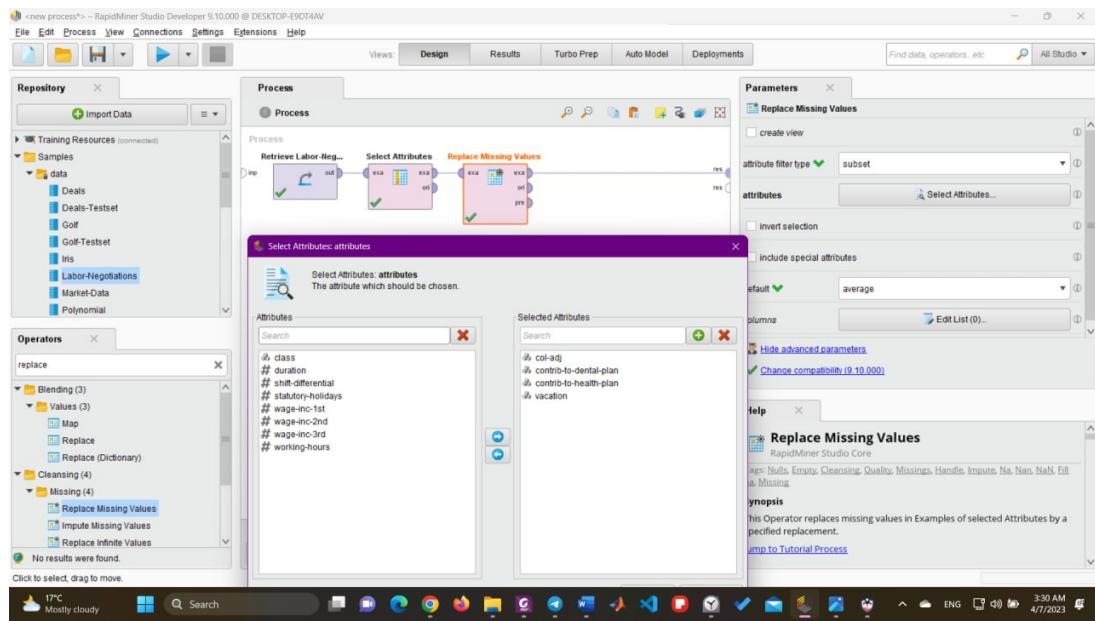
### مرحله ۱. وارد کردن مجموعه داده در برنامه



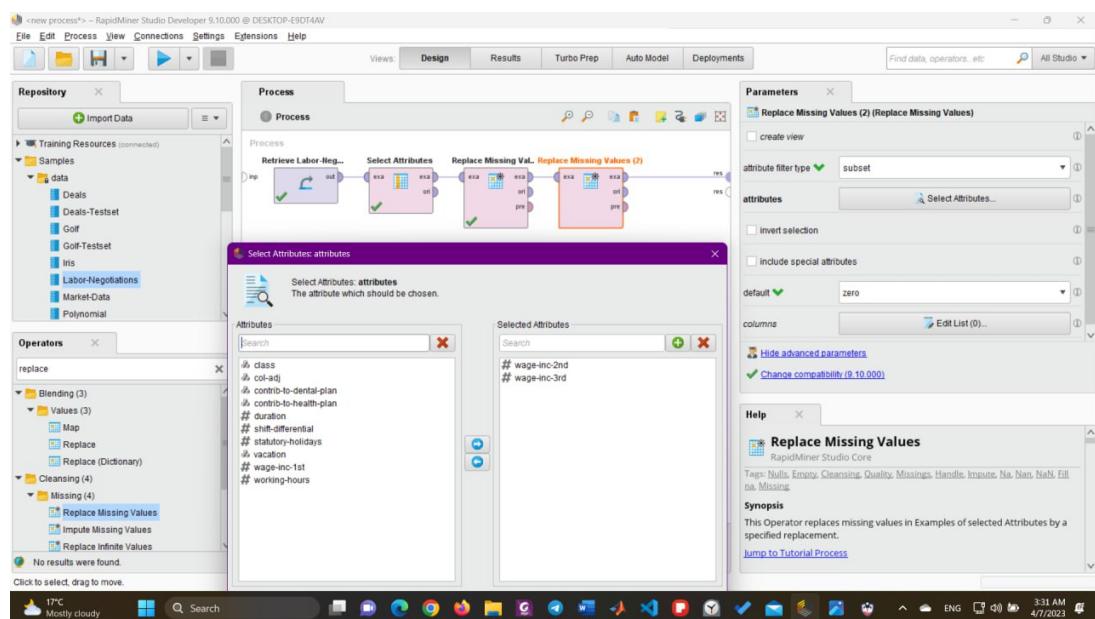
مرحله ۲. با استفاده از اپراتور Select Attributes ستونهایی که در بیش از ۲۰ رکورد مقدارشون miss شده بود را حذف کردیم و بقیه رو select کردیم. (چرا که ما کلا ۴۰ رکورد داریم و ستونی که در بیش از ۲۰ رکورد مقداری نداشته باشد به کار نخواهد آمد).



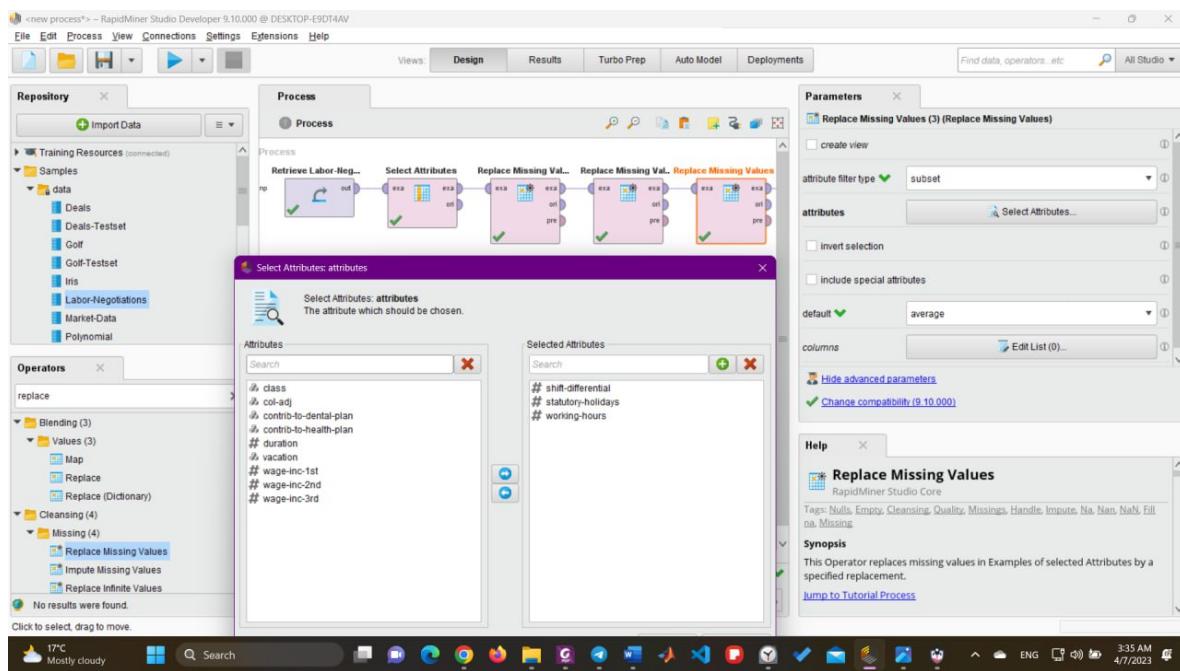
مرحله ۳. در این مرحله با استفاده از اپراتور Replace Missing Values مقادیر از دست رفته برای چهار بهنام‌های vacation, col-adj, contrib-to-dental-plan, contrib-to-health-plan attribute را با مقدار جایگزین کردیم. توجه شود که این هستند categorical attribute از جنس attribute هستند و میانگین average بنابراین برنامه به طور خودکار مُد آن‌ها را جایگزین می‌کند که مطابق خواسته ما هم است.



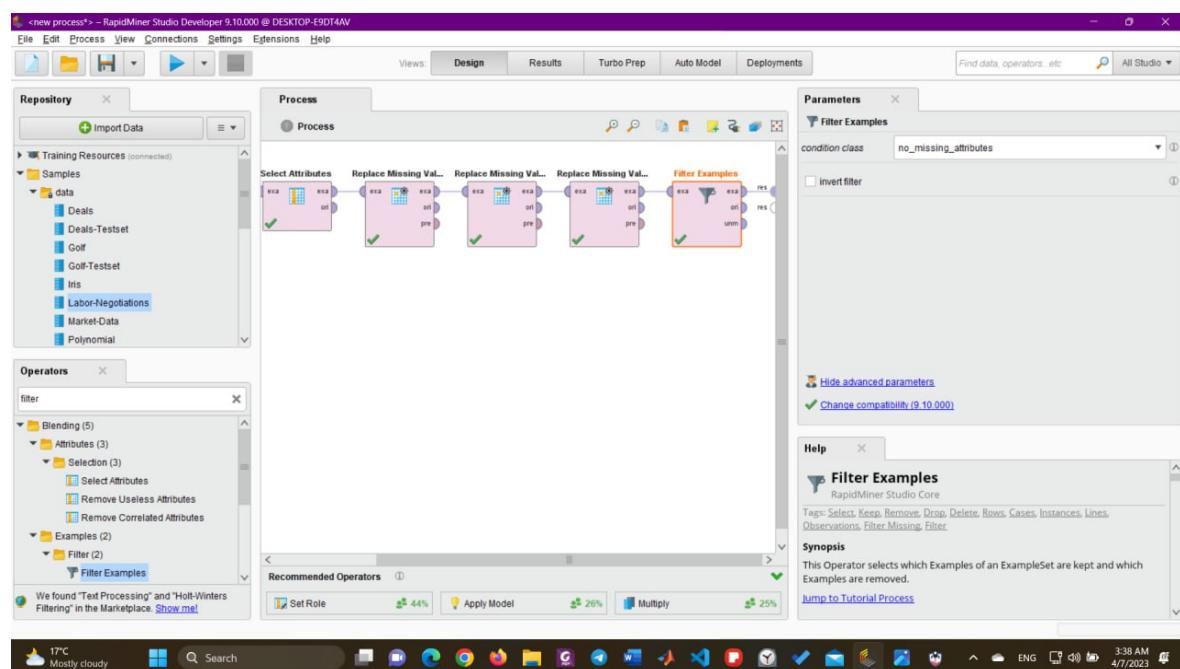
مرحله ۴. در این مرحله مقادیر از دست رفته برای wage-inc-3rd و wage-inc-2nd attribute را با مقدار صفر یا zero جایگزین کردیم. چرا که این دو فیلد مقدار افزایش دستمزد کارکنان بعد از دوسال و سه‌سال اشتغال را نشان می‌دهد و ممکن است که افرادی به این اندازه مشغول به کار نبوده باشند و این مقدار برای آن‌ها صفر باشد.



مرحله ۵. در این مرحله مقادیر از دسترفته سه ویژگی shift-differential, statutory-holidays را با مقدار میانگین آنها جایگزین کردیم، چرا که از نوع متغیر عددی ya working-hours بودند.

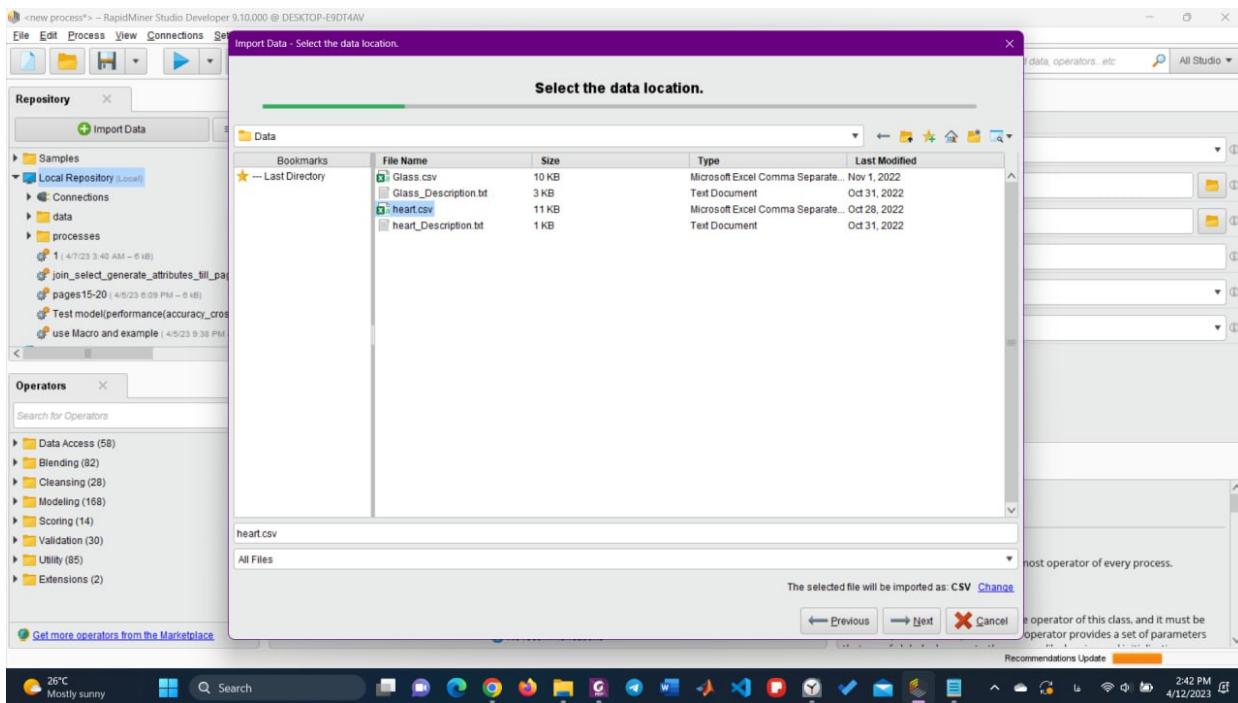
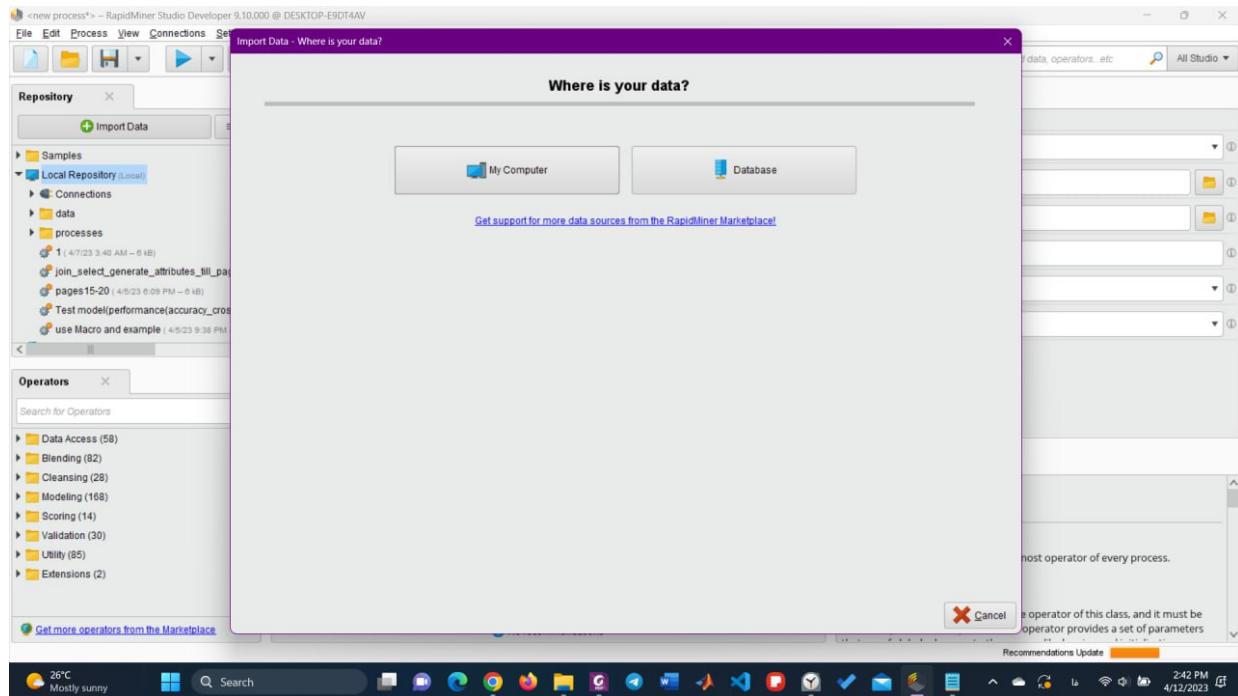


مرحله ۶. در این مرحله با استفاده از اپراتور Filter Examples و پارامتر no\_missing\_attributes محدود رکوردهایی که با وجود گذر از مراحل قبل هنوز بعضی از فیلدهایشان خالی مانده بود را از مجموعه داده‌مان خارج کردیم. چرا که تعداد ویژگی‌های از دسترفته‌شان بیشتر از باقی رکوردها بوده و با توجه به تعداد کمشان نمی‌ارزید که برای این‌کار تغییری در داده‌هایمان ایجاد کنیم.

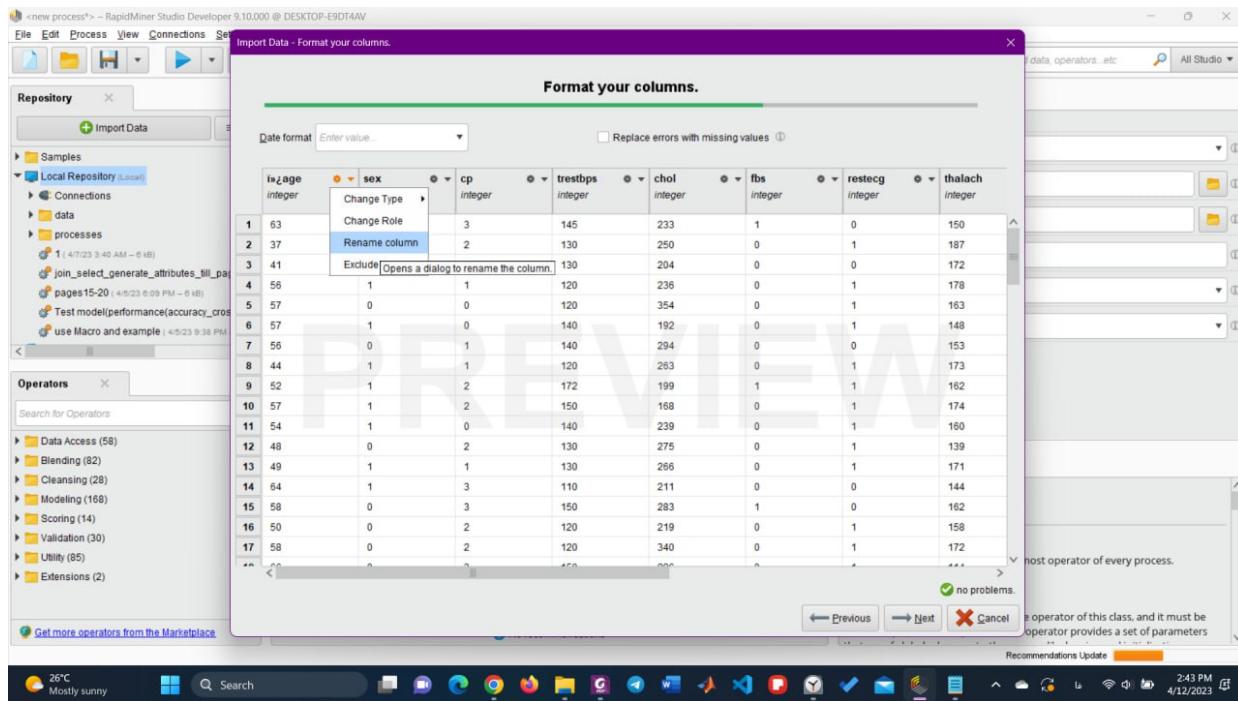


## بخش دوم. ۱ - طبقه‌بندی چندکلاسه و طبقه‌بندی درخت تصمیم (مجموعه‌داده Heart)

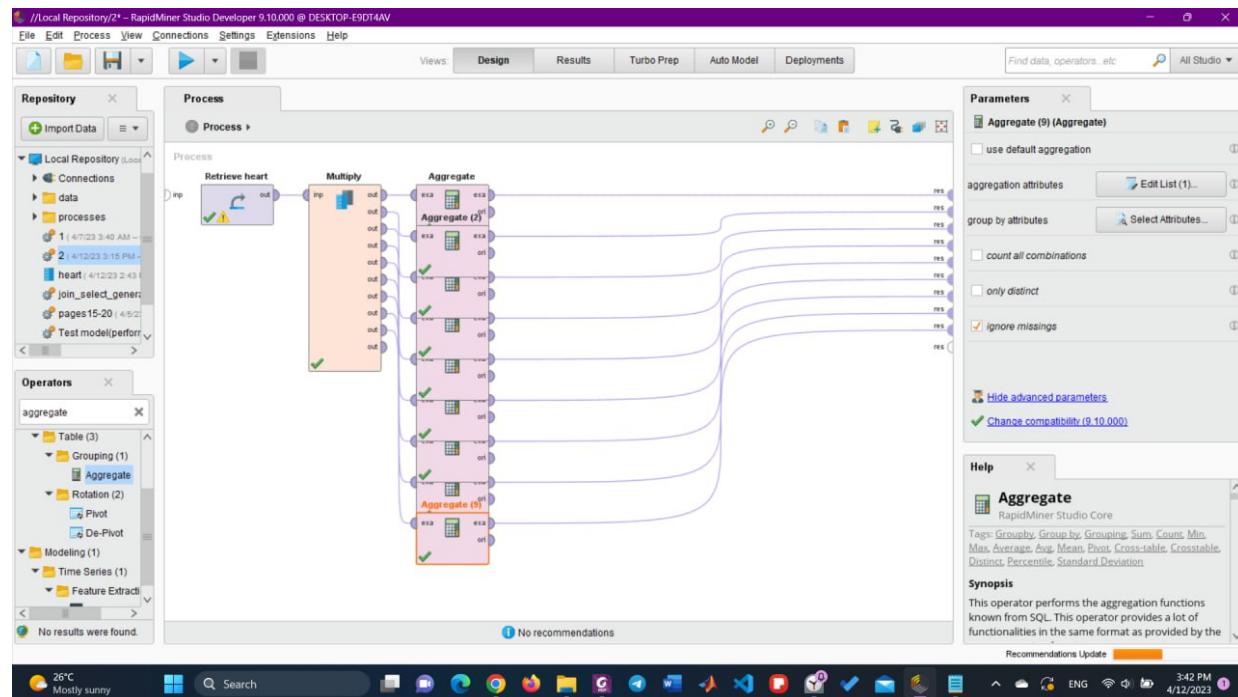
### مرحله ۱. ورود مجموعه‌داده داخل برنامه



## مرحله ۲. تغییر نام و clean کردن attribute مربوط به سن (age)



مرحله ۳. در این مرحله بهارای هر attribute میکنیم تا توسط Aggregate operator استفاده شود. بفهمیم که از هر یک از attribute های value چه تعداد داریم.



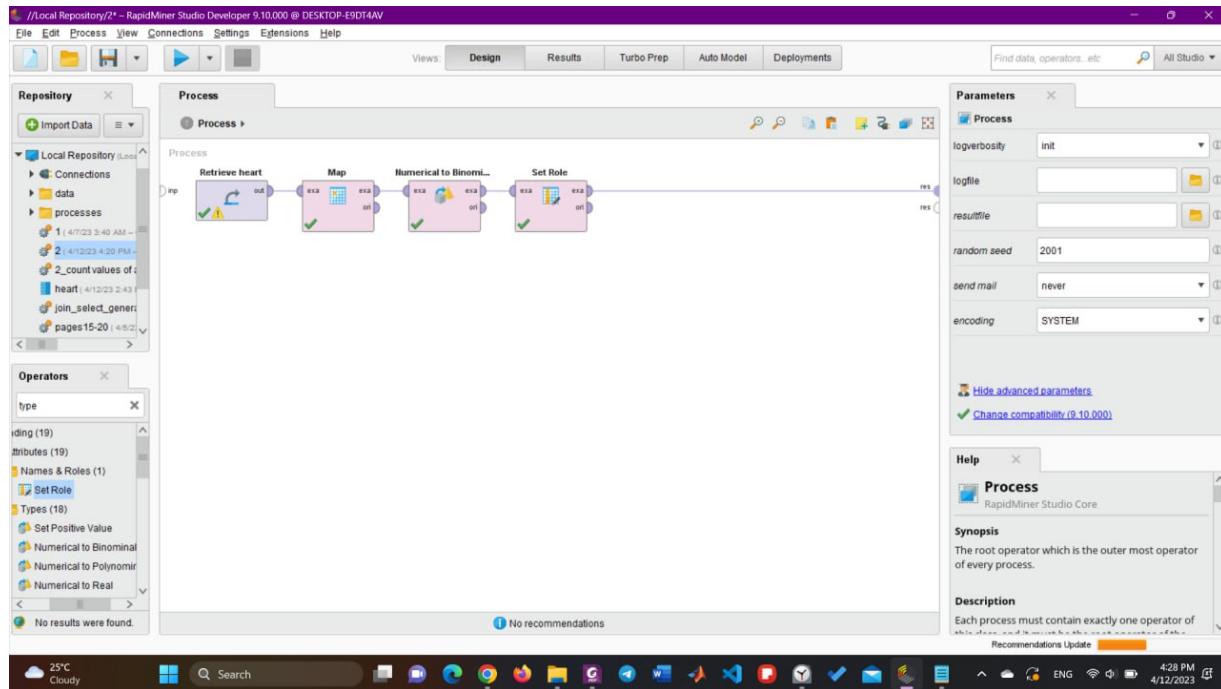
خب همانطور که در تصویر زیر مشاهده می‌کنید، برای ویژگی `thal` که طبق توضیحات نباید مقدار صفر داشته باشد، تعداد ۲ رکورد با مقدار صفر وجود دارد که این اشتباه است؛ برای تصحیح این دو مقدار را توسط عملگر `Map` با مقدار مُد که عدد ۲ است جایگزین می‌کنیم.

همچنین تصویر زیر که مربوط به `statistics` داده‌ها است نشان می‌دهد که اولاً هیچ مقدار از دست رفته‌ای نداریم و ثانیاً مقادیر مینیمم بزرگتر مساوی صفر هستند (منفی نیستند) پس دیتای نادرست هم نداریم.

همچنین توجه شود که عملگر `Aggregate` در تصویر بالا صرفا برای اطمینان از اعمال شدن تصحیحات بوده است.

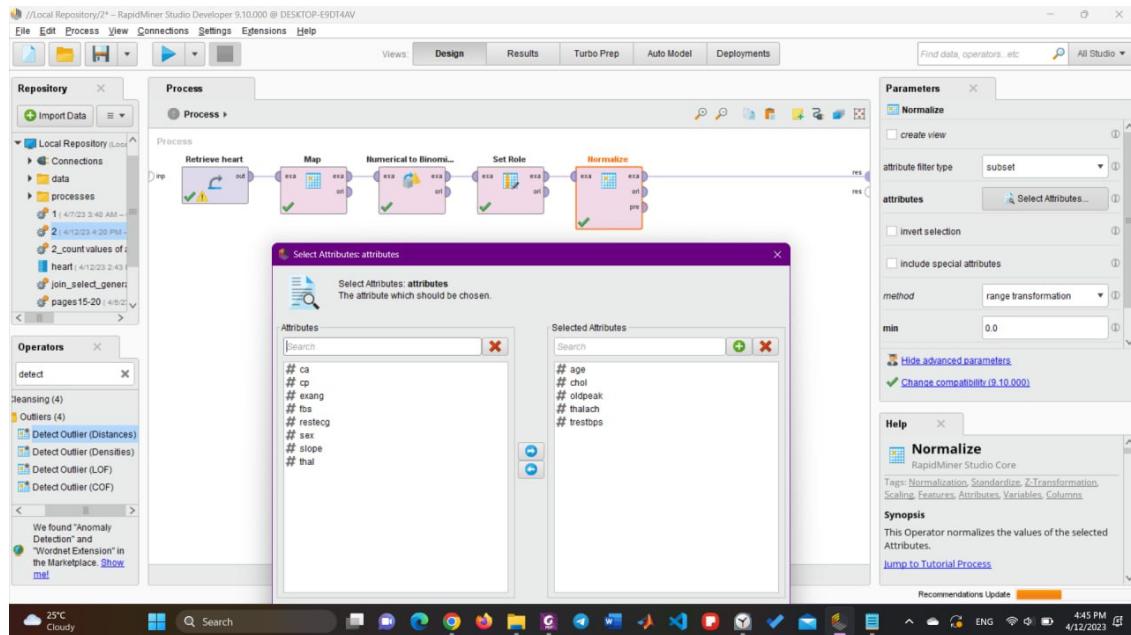
مرحله ۴. در این مرحله ابتدا با عملگر Numerical to Binominal مقادیر target را که متشکل از ۰ و ۱ هستند را به True و False تبدیل می‌کنیم تا بتوانیم آنرا به عنوان لیبل در نظر بگیریم.

سپس با استفاده از عملگر Set Role نقش label را به نام target می‌دهیم و همانطور که در تصویر دوم مشاهده می‌کنید رنگ مقادیر مربوط به این سبز می‌شود.

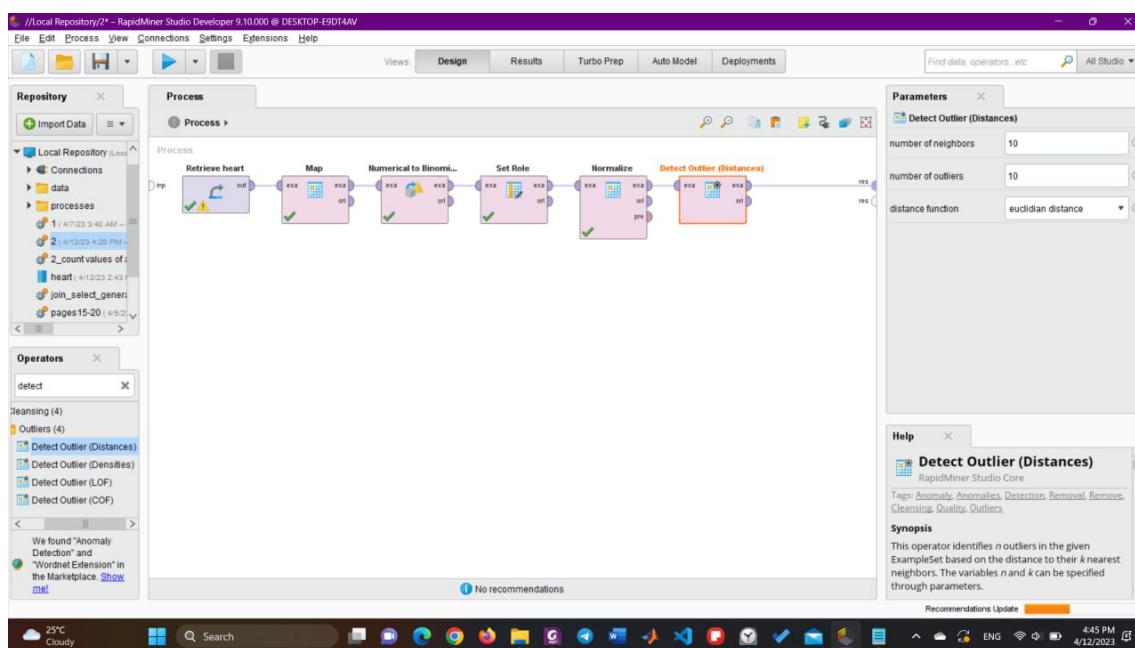


Row No.	target	thal	age	sex	cp	trestbps	chol	fbs	restecg	thal	thalab
156	true	2	58	0	0	130	197	0	1	131	
157	true	2	47	1	2	130	253	0	1	179	
158	true	2	35	1	1	122	192	0	1	174	
159	true	3	58	1	1	125	220	0	1	144	
160	true	3	56	1	1	130	221	0	0	163	
161	true	2	56	1	1	120	240	0	1	169	
162	true	2	55	0	1	132	342	0	1	166	
163	true	2	41	1	1	120	157	0	1	182	
164	true	2	38	1	2	138	175	0	1	173	
165	true	2	38	1	2	138	175	0	1	173	
166	false	2	67	1	0	160	286	0	0	108	
167	false	3	67	1	0	120	229	0	0	129	
168	false	2	62	0	0	140	268	0	0	160	
169	false	3	63	1	0	130	254	0	0	147	
170	false	3	53	1	0	140	203	1	0	155	
171	false	1	56	1	2	130	256	1	0	142	
172	false	3	48	1	1	110	229	0	1	168	
173	false	2	58	1	1	120	284	0	0	160	

مرحله ۵. در این مرحله با استفاده از عملگر Normalize ویژگی‌های attribute که عددی و غیر گسسته هستند را نرمال کرده‌ایم تا بتوانیم داده‌های پرت را در آنها شناسایی کنیم.

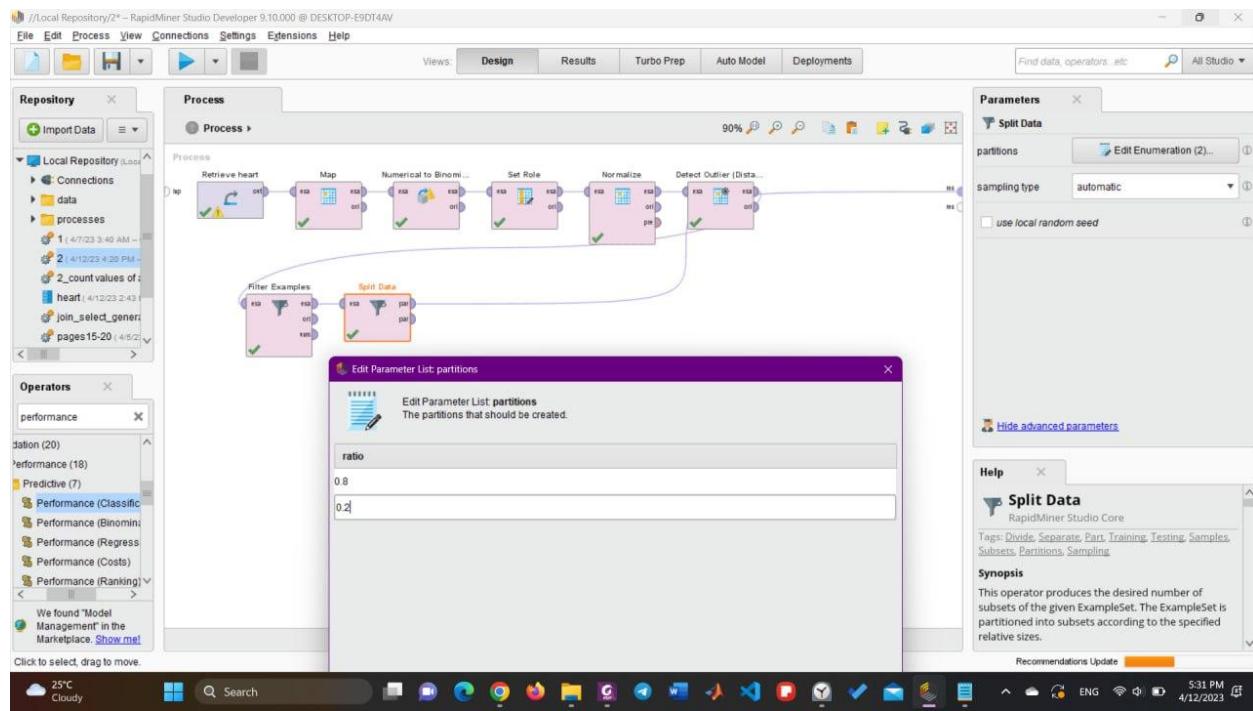


مرحله ۶. در اینجا با استفاده از عملگر Detect Outlier مقادیری که داده پرت محسوب می‌شوند را شناسایی می‌کنیم. با این‌کار یک فیلد به مجموعه‌اده اضافه می‌شود که مقدار true یا false دارد و نشان می‌دهد که آیا رکورد مدنظر دارای داده پرت است یا خیر. توجه شود که داده‌های پرت در عملگر مدنظر به طور پیش‌فرض روی عدد ۱۰ بوده که آن را تغییر ندادیم.

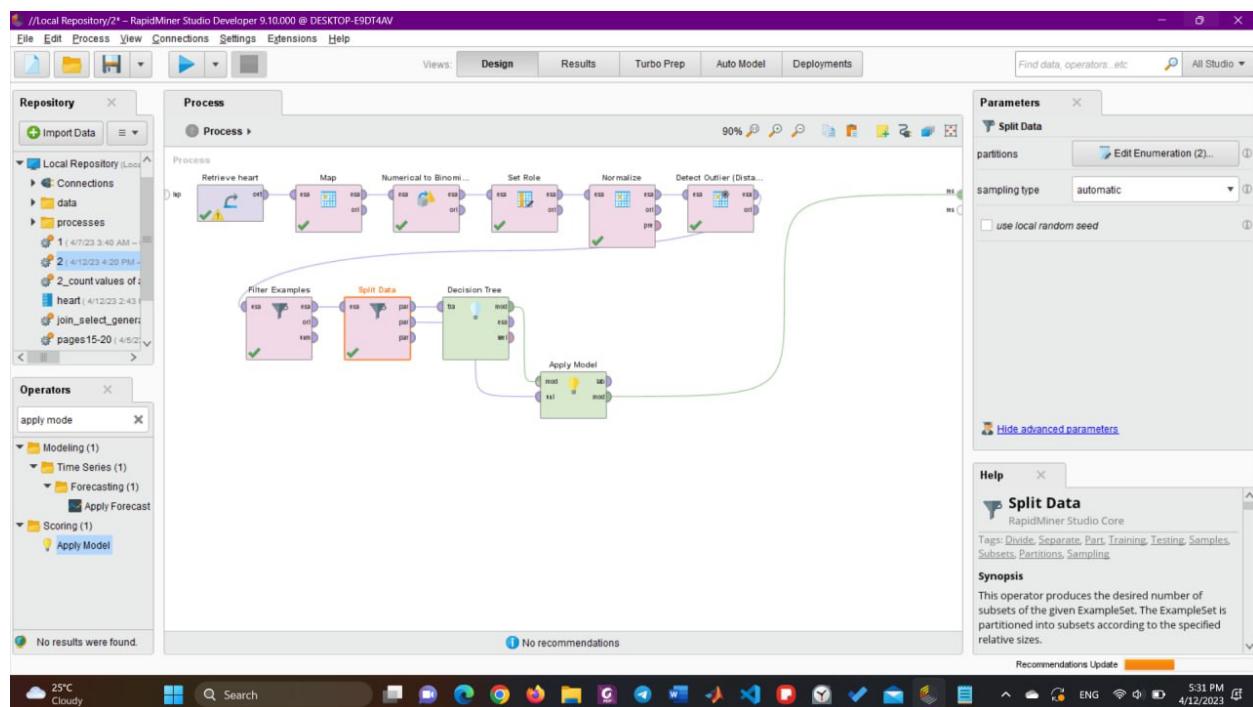


مرحله ۷. با استفاده از عملگر Filter Examples رکوردهایی که ویژگی outlier آنها false بوده را فیلتر می‌کنیم و با این کار بقیه موارد که دارای داده پرت هستند را از مجموعه داده‌مان خارج می‌کنیم.

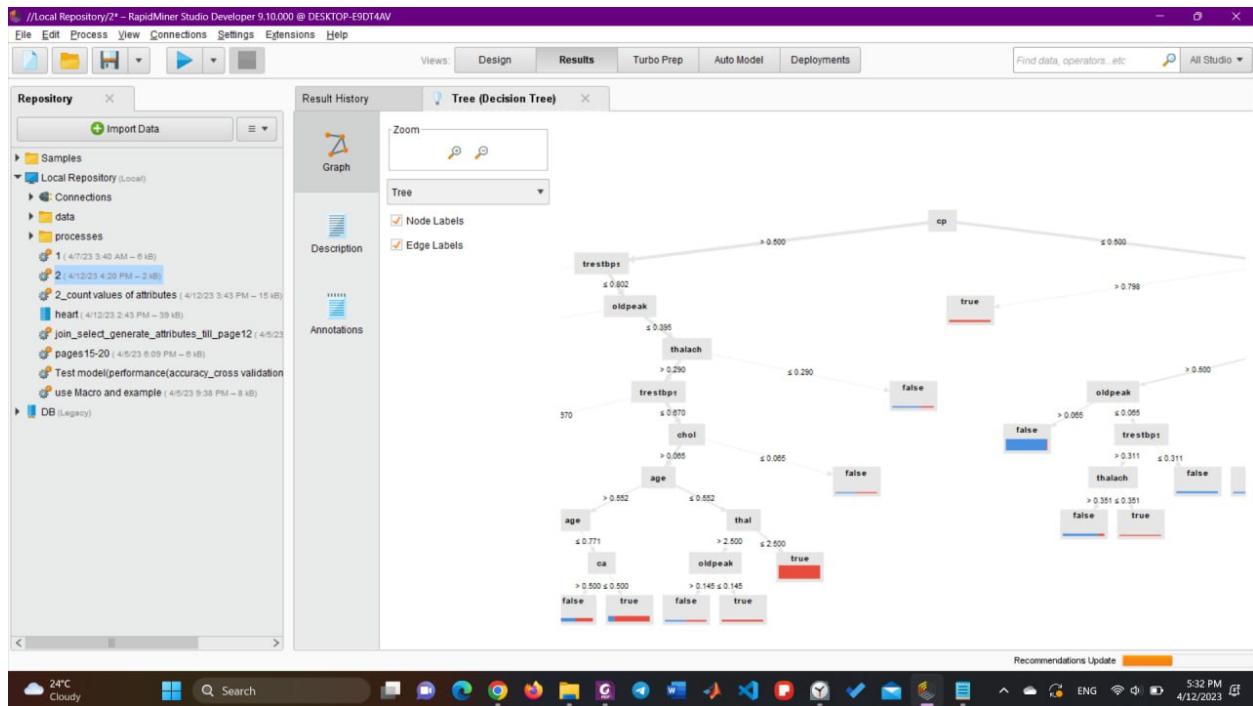
مرحله ۸. با استفاده از عملگر Split Data داده‌ها را با نسبت ۰.۲ و ۰.۸ تقسیم می‌کنیم.



**مرحله ۹.** اولین خروجی Decision Tree را که همان ۸۰ درصد داده ها است وارد Split Data کرده و سپس مدل درخت تصمیم حاصل را به همراه دومین خروجی Split Data (یا همان ۲۰ درصد داده ها) به عملگر Apply Model میدهیم.



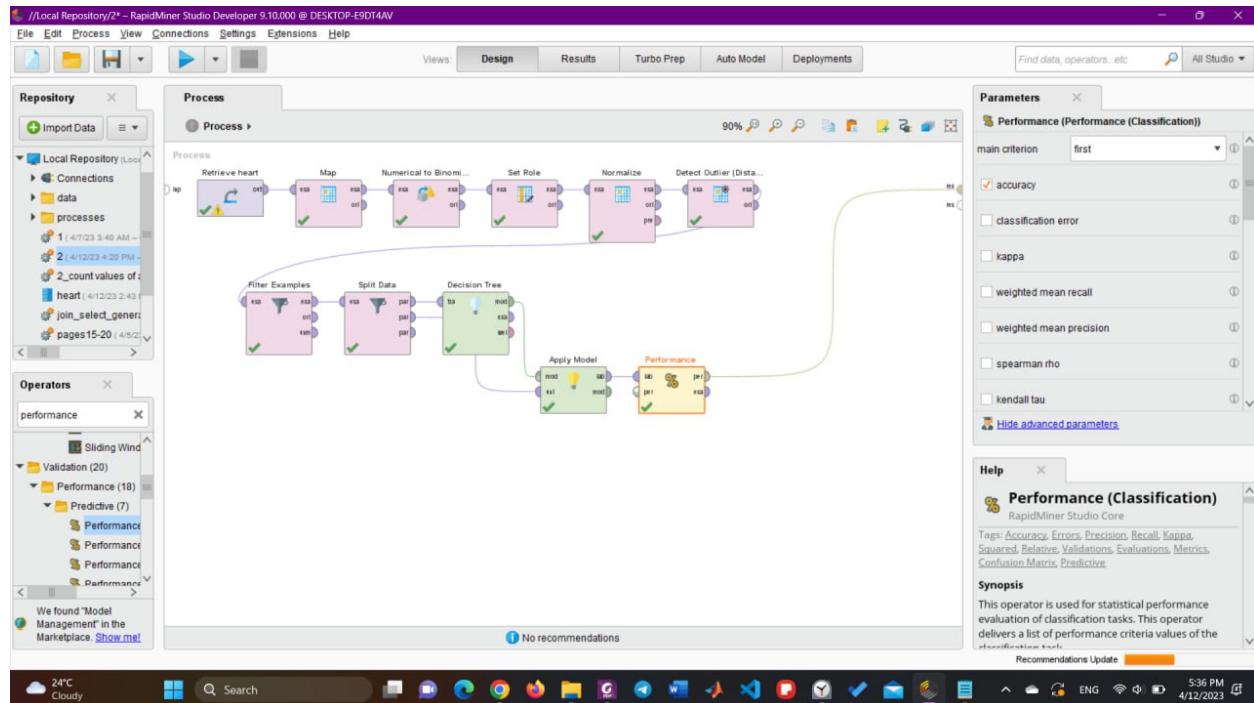
مرحله ۱۰. تصویر درخت تصمیم حاصل از داده‌ها را در زیر می‌بینید.



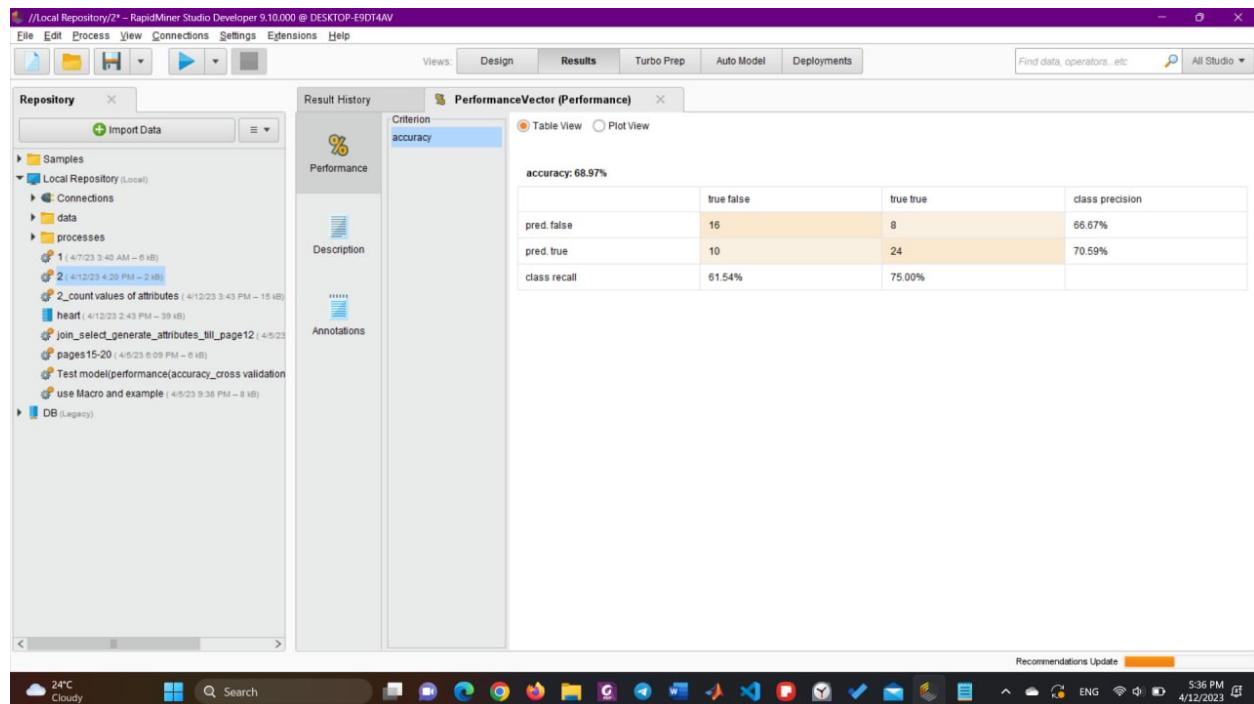
مرحله ۱۱. تصویر نتیجه اعمال مدل بر داده‌های آزمایشی در زیر آمده است. در این تصویر ستون های پیش بینی مقدار target و میزان قطعیت درباره مقدار آن و مقدار واقعی target قابل مشاهده است.

Row No.	target	prediction(t...)	confidence(t...)	confidence(t...)	outlier	age	trestbps	chol	thalach	oldpe...
1	true	false	0.978	0.022	false	0.583	0.245	0.521	0.702	0.097
2	true	true	0.143	0.857	false	0.562	0.434	0.384	0.626	0.210
3	true	true	0	1	false	0.312	0.245	0.313	0.779	0
4	true	false	0.500	0.500	false	0.479	0.736	0.167	0.695	0.081
5	true	true	0.143	0.857	false	0.583	0.528	0.096	0.786	0.258
6	true	false	1	0	false	0.292	0.528	0.276	0.763	0.242
7	true	true	0	1	false	0.625	0.387	0.247	0.687	0.081
8	true	true	0	1	false	0.312	0.340	0.244	0.824	0.065
9	true	true	0	1	false	0.521	0.387	0.406	0.756	0
10	true	true	0.143	0.857	false	0.750	0.623	0.534	0.611	0.129
11	true	true	0	1	false	0.500	0.321	0.205	0.336	0
12	true	false	0.500	0.500	false	0.312	0.132	0.034	0.794	0.097
13	true	true	0	1	false	0.333	0.340	0.247	0.794	0.097
14	true	true	0	1	false	0.312	0.245	0.215	0.756	0
15	true	true	0	1	false	0.250	0.170	0.283	0.824	0
16	true	false	0.500	0.500	false	0.479	0.547	0.393	0.817	0.194
17	true	false	0.857	0.143	false	0.583	0.358	0.185	0.740	0

## مرحله ۱۲. خروجي Apply Model را به عملگر Performance ميدهيم.



مرحله ۱۳. عملگر Performance دو خروجي جدول و نمودار دارد که در زير خروجي جدول آن نشان داده شده است.



در جدول عملکرد هر ردیف تعداد برچسب های True و False پیش بینی شده و هر ستون تعداد واقعی برچسبها را نشان میدهد. برای مثال در این تصویر مدل ما ۲۴ نتیجه False داشته که از این تعداد ۱۶ تای آنها مقدار واقعی شان False بوده و ۸ تا مقدار واقعی True داشته اند بنابراین دقت مدل برای این برچسب ۶۶٪ را ارزیابی شده است.

مقدار class recall برابر حاصل تقسیم تعداد پیش بینی های درست در هر ستون بر کل برچسب های آن ستون است.

مقدار class precision برابر حاصل تقسیم تعداد پیش بینی های درست مدل در هر برچسب (ردیف) تقسیم بر تعداد کل پیش بینی هایی که با آن برچسب داشته (جمع مقادیر ردیف) است.

در بالای جدول نیز دقت کلی مدل نشان داده شده است که با فرمول زیر محاسبه میشود:

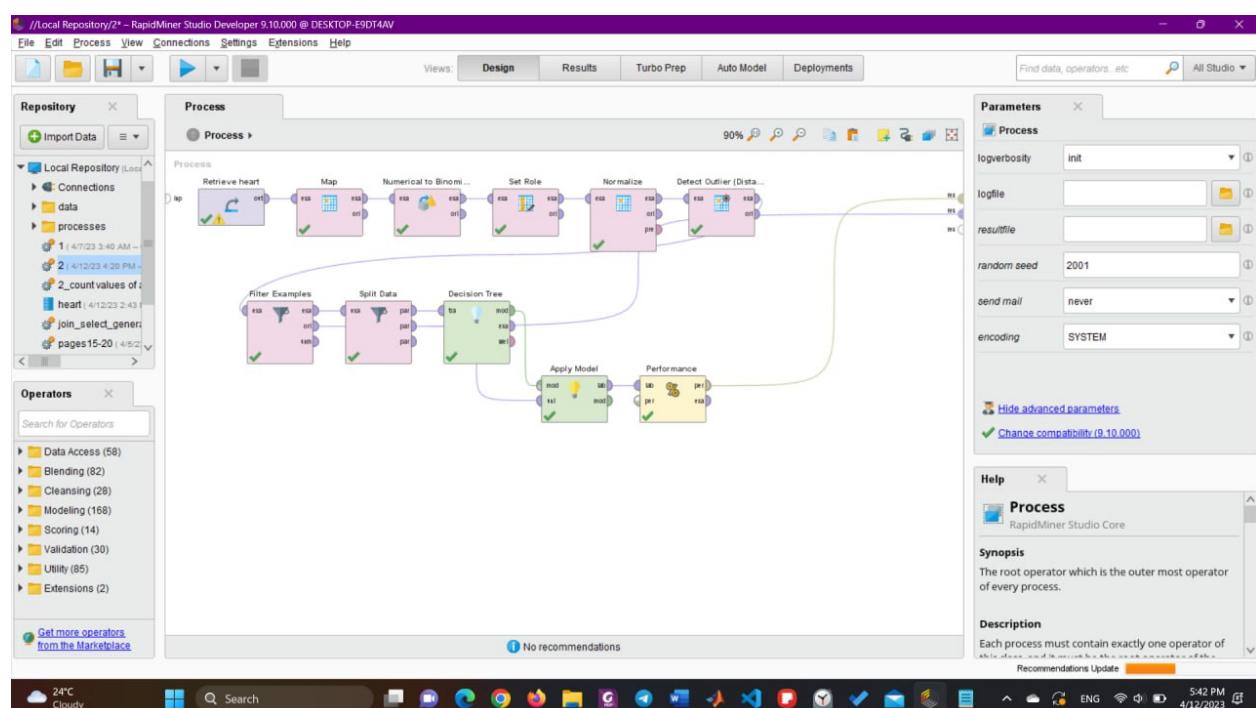
$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

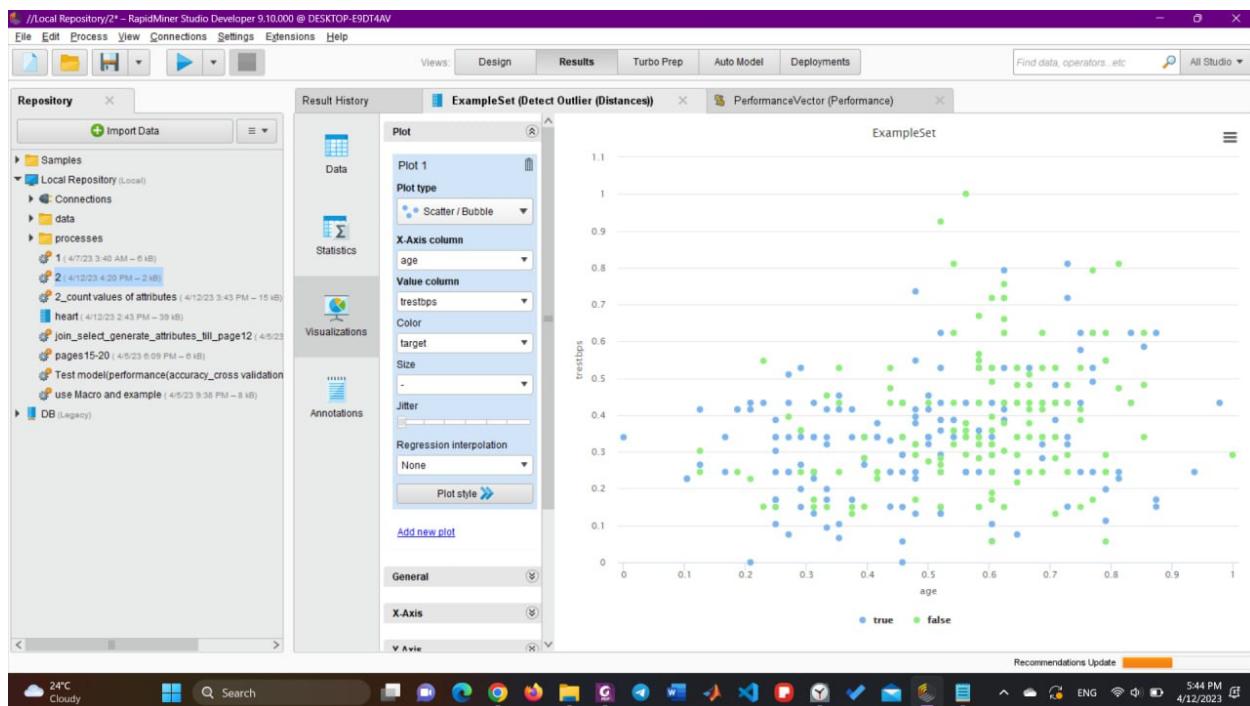
فرمول بالا تعداد پیش بینی های صحیح را برای تعداد کل پیش بینی ها تقسیم میکند.

در نمودار عملکرد، محور X مقدار واقعی برچسب ها، محور Y مقدار پیش بینی شده برچسب ها و محور Z تعداد را نشان میدهند به گونه ای که برای مثال ارتفاع میله در مختصات  $x=1$  و  $y=1$  تعداد داده هایی را نشان میدهد که مدل درخت تصمیم برچسب true به آنها نسبت داده و برچسب واقعی آنها نیز true بوده است.

## بخش دوم. ۲ - نمودارهای scatter plot (مجموعه داده Heart)

در این بخش داده هارا به طور مستقیم به خروجی برنامه متصل کرده و سپس نمودار های پراکندگی را برای آنها رسم میکنیم.

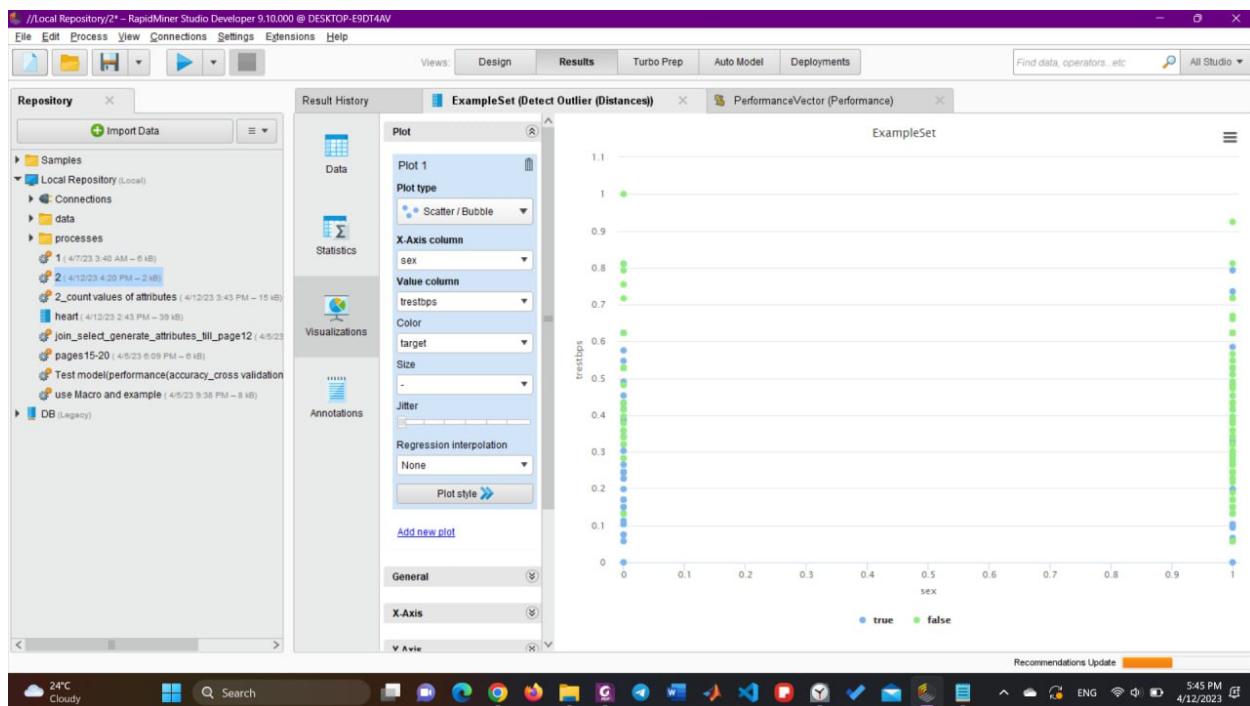




در نمودار بالا محور افقی ویژگی سن و محور عمودی ویژگی فشار خون افراد در حالت نشسته یا استراحت انتخاب شده و رنگ نمودار هم بر اساس برچسب بیماری قلبی تعیین میشود.

پراکندگی داده ها در این نمودار اطلاعات زیادی به دست نمیدهد اما میتوان برای مثال به دو نتیجه گیری از این نمودار اشاره کرد:

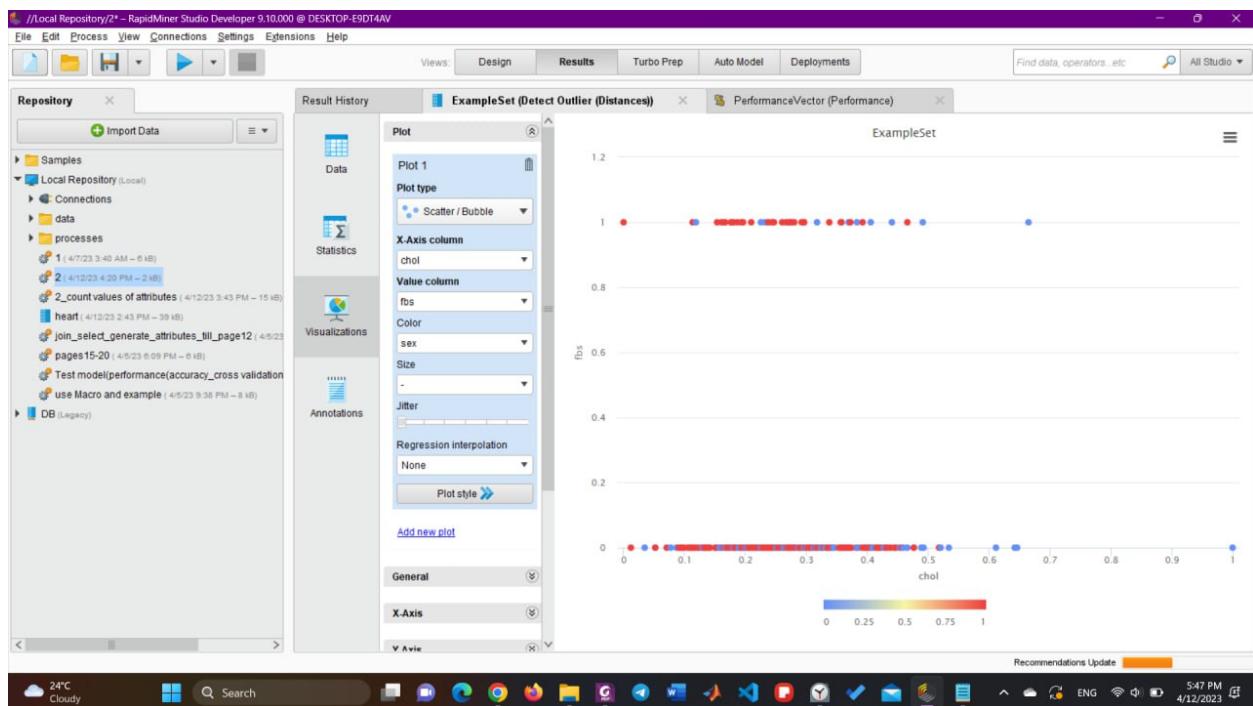
- افراد با بیماری های قلبی بیشتر بین سنین 40 تا 60 و افراد بدون بیماری بین سنین 50 تا 70 قرار داشتند. یا به طور کلی افراد بدون بیماری قلبی سن بیشتری نسبت به افراد با بیماری قلبی داشتند.
- افرادی بدون بیماری قلبی با فشار خون بالا تر از مقدار نرمال شده 0.8 در داده ها وجود دارند اما هیچ فردی با بیماری قلبی فشار خونی بالاتر از این مقدار نداشته، که این میتواند نشاندهنده کنترل بیشتر فشار خون توسط افراد با بیماری قلبی یا خطر مرگ بیشتر فشار خون بالا برای افراد با بیماری قلبی باشد.



تصویر بالا نمودار پراکندگی با مقدار جنسیت برای محور افقی و فشار خون در حالت نشسته برای محور عمودی را نشان میدهد، رنگ نمودار مانند حالت قبل همان ویژگی هدف یا بیماری قلبی است. مقدار 1 برای جنسیت معرف مرد و مقدار 0 معرف زن است.

براساس این نمودار میتوان نتیجه گیری های زیر را بیان کرد:

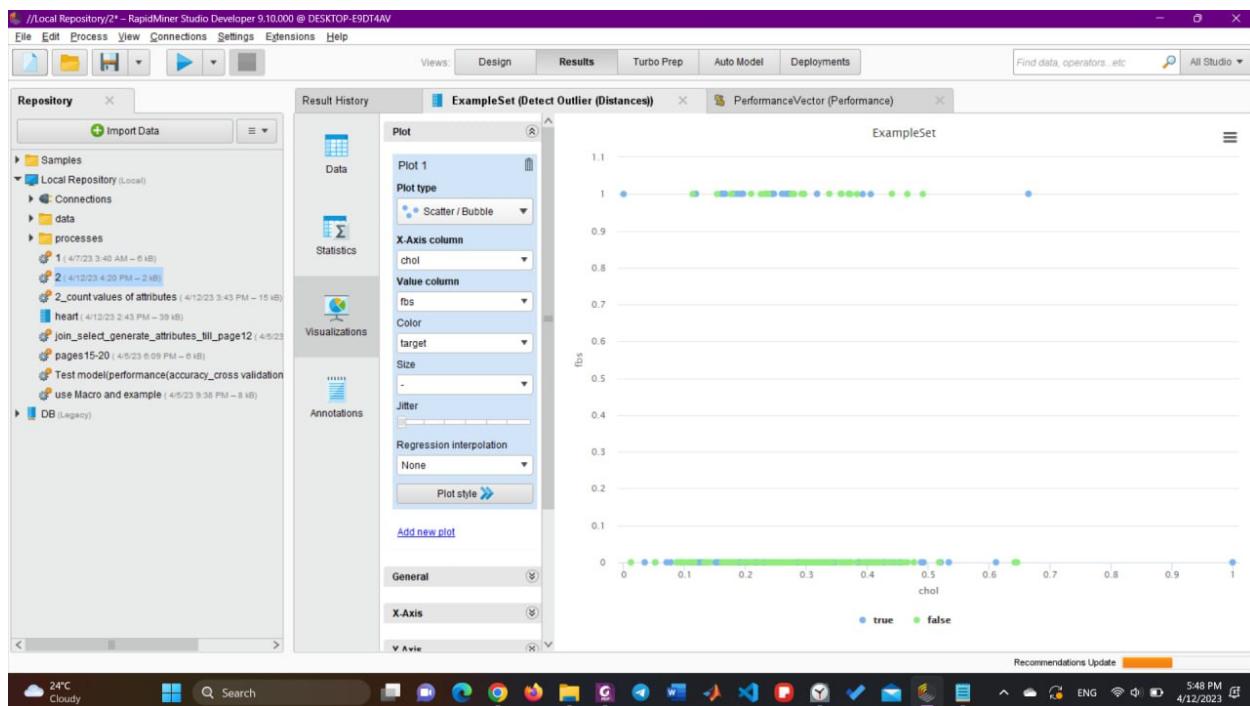
- مردان با بیماری قلبی به طور میانگین فشار خون بالاتری نسبت به زنان با بیماری قلبی داشته‌اند.
- پراکندگی فشار خون در مردان بدون بیماری قلبی از زنان این گروه بیشتر بوده و فشار خون زنان بدون بیماری قلبی بیشتر مقادیری اطراف مقدار نرمال شده 0.4 داشته است.
- فشار خون زنان با بیماری قلبی پراکندگی بیشتری نسبت به فشار خون زنان بدون بیماری قلبی دارد و این مورد برای مردان هم در نمودار دیده میشود.



در نمودار بالا محور افقی نشان دهنده میزان کلسترول خون و محور عمودی بیان کننده این است که آیا قند خون فرد در حالت ناشتا مقداری بیشتر از 120 دارد(1) یا خیر(0). جنسیت نیز به عنوان رنگ نمودار انتخاب میشود.

همانطور که میبینید چون ویژگی جنسیت به صورت عددی تعریف شده و نه به صورت باینری، نمودار برای آن طیف رنگ در نظر گرفته. با توجه به نمودار بالا:

- تعداد مردان با قند خون بالا بیشتر از زنان است.
- زنان به طور میانگین کلسترول بیشتری نسبت به مردان داشته اند.
- پراکندگی کلسترول خون در افراد با قند خون نرمال بیشتر از افراد با قند خون بالا است



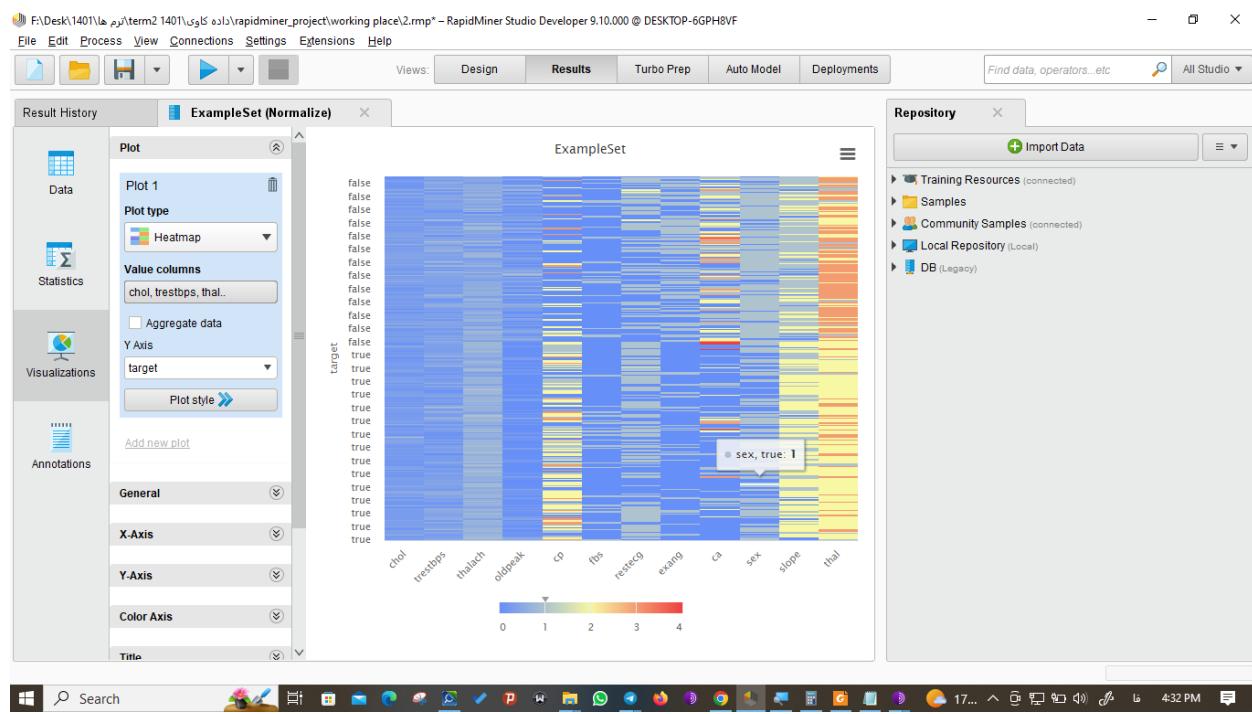
نمودار بالا همانند نمودار قبلی است با این تفاوت که در این نمودار به جای جنسیت، ویژگی هدف یا بیماری قلبی به عنوان رنگ نمودار انتخاب شده است. با توجه به این نمودار میتوان نتایج زیر را بیان کرد:

- افراد با بیماری قلبی کلسترول خون محدود تر و پایین تری نسبت به افراد بدون بیماری قلبی دارند با این وجود بیشترین کلسترول خون در کل داده ها مربوط به فردی با بیماری قلبی بوده است.
- افراد با قند خون بالا تعداد کمتری نسبت به افراد با قند خون نرمال دارند اما رابطه مشخصی میان قند خون بالا و بیماری قلبی در این نمودار دیده نمیشود.

## بخش دوم. ۳ - نمودار Heat Map (مجموعه داده Heart)

برای رسم نمودار Heat Map همه ویژگی ها به جز ویژگی سن را برای ستون ها و مقدار ویژگی هدف را برای محور y انتخاب میکنیم. دلیل اینکه ویژگی سن را به نمودار اضافه نمیکنیم اینست که این ویژگی دامنه تغییر بیشتری دارد و در صورت اضافه شدن پراکندگی رنگ بیشتری را در نمودار به خود اختصاص داده و بقیه ستون ها یکرنگ دیده خواهند شد.

تصویر نمودار حاصل را در زیر میبینید.



در این نمودار مقادیر true یا دارای بیماری قلبی همه پایین محور و مقادیر false بالای محور ع قرار دارند. با توجه به این نمودار به سادگی میتوان تشخیص داد که کدام ویژگی ها تاثیر محسوسی بر ویژگی هدف دارند.

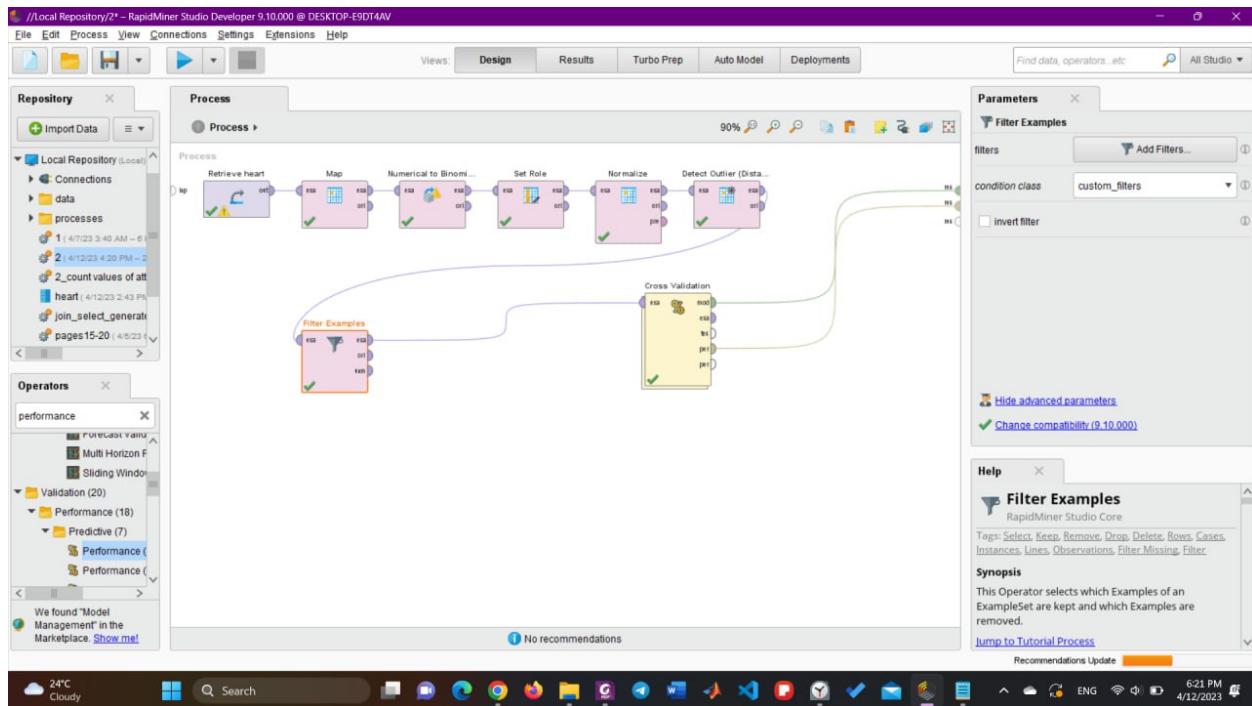
این ویژگی ها در زیر آمده اند:

- ویژگی thal یا داشتن بیماری تالاسمی
- ویژگی slope
- ویژگی sex
- ویژگی ca
- ویژگی exang
- ویژگی cp

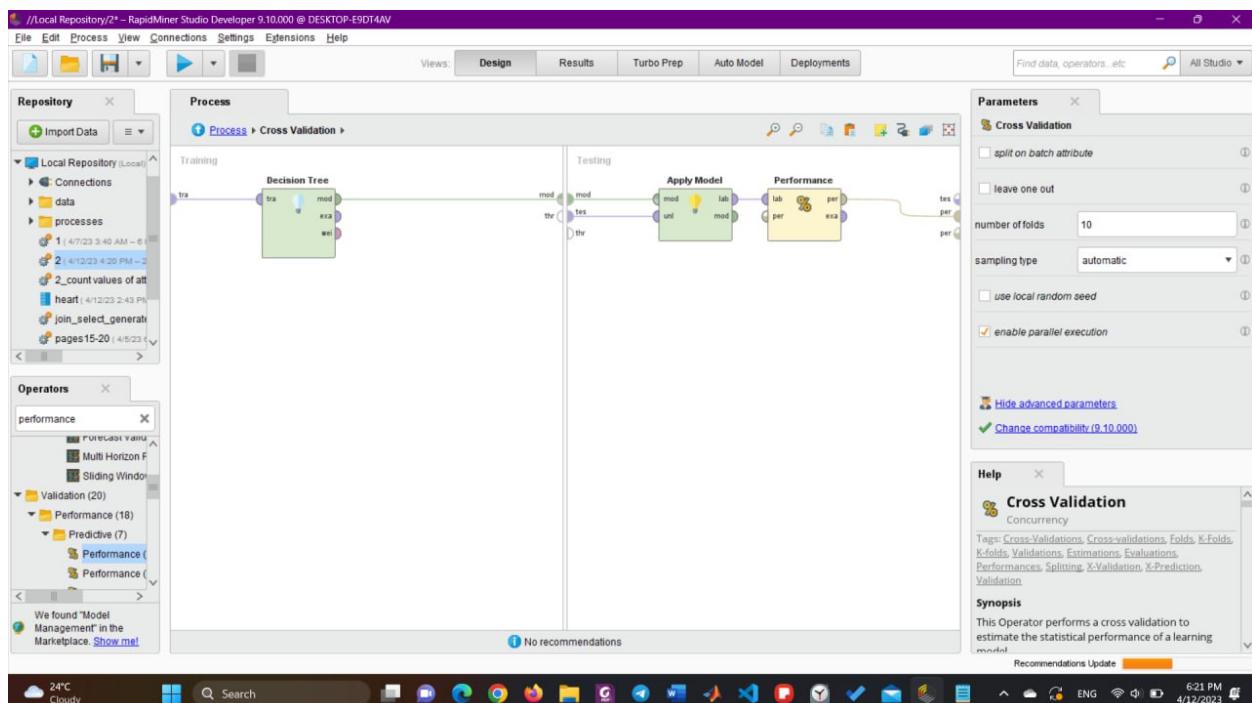
در ستون مربوط به هرکدام از این ویژگی ها رنگ نمودار از بالا به پایین تغییر زیاد و مشخصی داشته که نشاندهنده ارتباط بیشتر این ویژگی ها با ویژگی هدف یا بیماری قلبی است.

## بخش دوم. ۴ (Heart) مجموعه داده Cross Validation

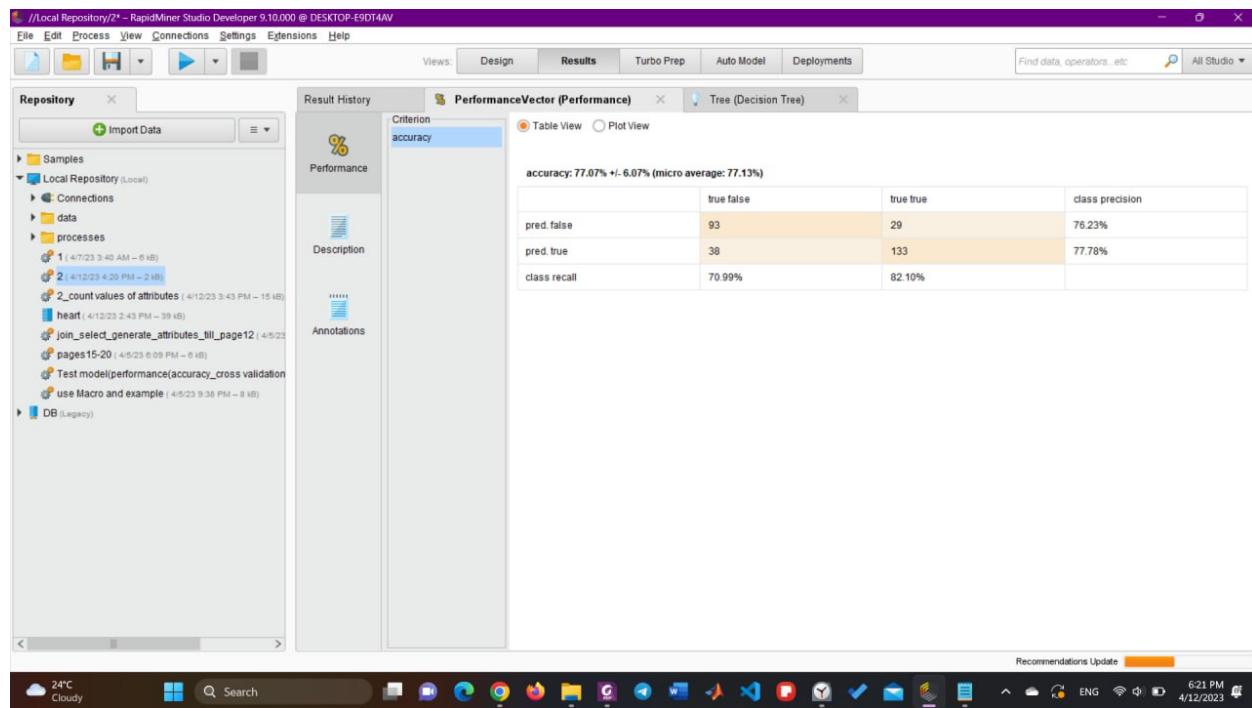
مرحله 1. در این بخش عملگر Cross Validation را به فرآیند اضافه کرده و خروجی های مدل و ارزیابی آن را به خروجی برنامه متصل میکنیم. مقدار number of folds (10) قرار میدهیم.



مرحله 2. در قسمت آموزش در بخش Cross Validation مربوط به مدل درخت تصمیم و در قسمت آزمایش آن عملگر های Apply Model و Performance را وارد میکنیم.



### مرحله 3. با اجرای فرآیند مدل و ارزیابی آن بدست می‌آید



در قسمت ارزیابی accuracy نمایش داده شده برابر میانگین مقدار accuracy مدل ها بعلاوه انحراف معیار آنهاست.

در تصویر بعد مدل درخت تصمیم حاصل را میبینید:

